

## **LAPORAN PENELITIAN**

**Judul :**

### **IMPLEMENTASI CIRCULAR LINKED LIST PADA PEMBUATAN GAME PERMAINAN TRADISIONAL CONGKLAK**



**Oleh :**

**Agung Budi Prasetyo, S.Kom,M.Kom**

**NIDN : 0003087106,**

**NIP : 107108032005011001**

**Dilaksanakan Atas Bantuan Biaya Penelitian dari Puslitbang dan PPM**

**Semester Ganjil 2011/2012**

**Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM**

**YOGYAKARTA**

**2014**

## HALAMAN PENGESAHAN

Judul Penelitian : Implementasi Circular Linked List pada Pembuatan Game Permainan Tradisional Congklak

Bidang Ilmu : Pemrograman dan Rekayasa Perangkat Lunak

Ketua Peneliti :

a. Nama Lengkap : Agung Budi Prasetyo, S.Kom, M.Kom

b. NIDN : 0003087106

c. NPP/ NIP : 197108032005011001

d. Pangkat / Golongan : Penata Muda / IIIA

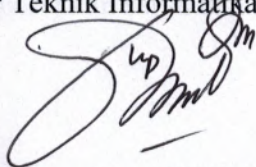
e. Jabatan Fungsional : Asisten Ahli

f. Jurusan/ Prodi : Teknik Informatika dan Komputer

g. Alamat Institusi : STMIK AKAKOM Yogyakarta, Jl.Raya Janti 143 Yogyakarta.

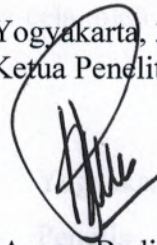
Waktu Penelitian : 6 bulan

Mengetahui  
Kajur Teknik Informatika



(Febri Nova Lenti, S.Si, M.T.)  
NIP/NIK 961079

Yogyakarta, 30 Oktober 2014,  
Ketua Peneliti,



(Agung Budi Prasetyo, S.Kom., M.Kom.)  
NIP/NIK 197108032005011001

Menyetujui,  
Kepala Puslitbang dan PPM



(Dra. Syamsu Windarti, M.T)  
NIP/NIK 1966071019932001



## KATA PENGANTAR

Puji syukur kepada Tuhan atas bimbingan dan kekuatan yang diberikanNya hingga penelitian ini dapat penulis selesaikan pada waktunya.

Pada kesempatan ini, penulis bermaksud mengucapkan terima kasih atas segala bantuan, bimbingan dan dukungan mulai dari masa persiapan, penggarapan hingga penulisan penelitian ini kepada :

1. Ketua STMIK AKAKOM Yogyakarta.
3. Kepala Puslitbang AKAKOM.
4. Staff Dosen AKAKOM yang telah membantu memberikan berbagai masukan.dalam pembuatan penelitian ini.

Tak ada gading yang tak retak, penulis menyadari sepenuhnya bahwa apa yang penulis susun dalam penelitian ini bukanlah tanpa cela, oleh karena itu penulis sangat mengharapkan saran dan masukan dari pembaca.

Yogyakarta, Oktober 2014

Penulis



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR .....	vi
DAFTAR TABEL .....	vi
DAFTAR LAMPIRAN .....	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian .....	3
1.7 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	4
BAB III LANDASAN TEORI.....	5
3.1 Permainan Congklak .....	5
3.1.1. Definisi .....	5
3.1.2. Bentuk Papan Congklak.....	6
3.1.3. Cara Memainkan Congklak .....	7
3.1.4. Flowchart Permainan Congklak .....	8
3.1.5. Permainan Congklak Digital .....	9
3.2 Circular Linked List .....	9
BAB IV ANALISA DAN PERANCANGAN.....	12
4.1 Spesifikasi Kebutuhan.....	12
4.2 Tahap Analisis dan Perancangan Sistem .....	12
Perancangan Algoritma dan Program Java.....	12
Perancangan Struktur Data.....	19
Perancangan Fungsi-fungsi .....	21
Perancangan Depth Limited Search (DLS) .....	22
Perancangan Antar Muka.....	23
BAB V IMPLEMENTASI DAN PEMBAHASAN.....	24
5.1 Implementasi.....	24
5.2 Pembahasan .....	24
BAB VI KESIMPULAN DAN SARAN.....	31
6.1 Kesimpulan.....	31
6.2 Saran.....	31



DAFTAR PUSTAKA.....	32
LAMPIRAN.....	34
LAMPIRAN 1 : RINCIAN BIAYA PENELITIAN .....	35
LAMPIRAN 2 : SK Ketua STMIK AKAKOM No: L.05.1 / 1551 / XI / 2011 .....	36

Lampiran 3.1	Beleidigende Congklak yang terdapat di dalam gambar	5
Lampiran 3.2	Pertemuan Congklak yang sedang di mainkan	6
Lampiran 3.3	Congklak dan Biji-bijinya	7
Lampiran 3.4	Pengertian Ruler & Cara Pemakaian Congklak	8
Lampiran 3.5	Contoh Diagram	10
Lampiran 3.6	Contoh Linkan dan cara Link	11
Lampiran 4.1	Beleidigende Congklak yang terdapat di dalam gambar	12
Lampiran 4.2	Pengertian Congklak dan Biji-bijinya	17
Lampiran 4.3	Diagram Congklak yang terdapat di dalam gambar	20
Lampiran 4.4	Diagram Congklak yang terdapat di dalam gambar	20
Lampiran 5.1	Interaksi Permainan Congklak Digital	34
Lampiran 5.2	Gambar Analisis Kemungkinan pada permainan Level	25
Lampiran 5.3	Peta Kemungkinan yang dihasilkan	27

#### DAFTAR TABEL

No	Keterangan	Hal
Tabel 3.1	Analisa langkah-langkah yang mungkin dilakukan	28
Tabel 3.2	Analisa kemungkinan untuk menang	29

#### DAFTAR LAMPIRAN

No	Keterangan	Hal
Lampiran 1	RINCIAN BIAYA PENELITIAN	35
Lampiran 2	SK Ketua STMIK AKAKOM No: L.05.1 / 1551 / XI / 2011	36



## DAFTAR GAMBAR

Nomor	Keterangan	Hal
Gambar 3.1	Bentuk Fisik Congklak yang terbuat dari kayu	5
Gambar 3.2	Permainan Congklak yang sedang dimainkan	6
Gambar 3.3	Congklak dan Bijinya	7
Gambar 3.4	Flowchart Rule of Game Permainan Congklak	9
Gambar 3.5	Circular Linked list	10
Gambar 3.6	Circular Linked list para Tree	11
Gambar 4.1	Flowchart dari fungsi untuk menjalankan 1 giliran	13
Gambar 4.2	Flowchart untuk menjalankan 1 genggam	17
Gambar 4.3	Skema simpul yang digunakan	20
Gambar 4.4	Skema simpul (Circular Linked list) pada aplikasi congklak	20
Gambar 5.1	Interface Permainan Congklak Digital	24
Gambar 5.2	Contoh Analisa Kemungkinan pada kedalaman 4 level	25
Gambar 5.3	Pohon Kemungkinan yang dihasilkan	27

## DAFTAR TABEL

Nomor	Keterangan	Hal
Tabel 5.1	Analisa langkah-langkah yang mungkin dilakukan	28
Tabel 5.2	Aspek kecukupan untuk menang	29

## DAFTAR LAMPIRAN

Nomor	Keterangan	Hal
Lampiran 1	RINCIAN BIAYA PENELITIAN	35
Lampiran 2	SK Ketua STMIK AKAKOM No: L.05.1 / 1551 / XI / 2011	36



## ABSTRAK

Teknik pengalokasian media penyimpan (memori) dengan menggunakan circular linkedlist secara teoritik menjanjikan kelenturan (dinamis) terhadap kebutuhan penggunaan memori. Hal ini karena linkedlist menerapkan konsep pengalokasian memori secara insidental (hanya saat dibutuhkan) dan pendealokasian memori saat memori tidak lagi digunakan. Konsep ini sangat sesuai jika diterapkan dalam pembuatan aplikasi di mana kebutuhan kapasitas memorinya tidak diketahui secara pasti (baru diketahui saat running).

Pada penelitian ini akan dilakukan percobaan pemanfaatan circular linkedlist untuk mengelola kebutuhan media penyimpan dari sebuah game bernama congklak. Game congklak adalah sebuah permainan yang memerlukan keahlian pemainnya dalam menentukan langkah yang paling tepat dari sekian langkah-langkah permainan yang ada. Kemampuan memilih langkah yang tepat akan sangat menentukan kemenangan pemain. Namun untuk dapat memilih langkah dengan tepat dibutuhkan suatu pohon keputusan (spanning tree) yang tentunya juga memerlukan media penyimpan yang besar namun luwes. Di sinilah akan dilihat apakah circular linkedlist mampu membantu mengatasi kebutuhan yang diminta.

Dalam sebuah purwarupa aplikasi berbasis sistem operasi windows yang dibangun menggunakan bahasa pemrograman java aplikasi ini diharapkan mampu memberikan keuntungan lebih dalam hal kelenturan penggunaan media penyimpan.

**Kata kunci :** Circular Linked list, congklak



## ABSTRACT

Memory allocation techniques using circular linkedlist theoretically promising more flexibility memory usage. It is because linkedlist apply the incidental memory allocation (only when needed) and deallocation when the memory is no longer used. This concept is appropriate when applied to develop an application in which the needs of its memory capacity is not known (only known when running).

In this research will be carried out experiments using circular linked list to manage the memory needs of a game called congklak. Congklak is a game that requires skill of players in determining the most appropriate step of steps there are. The ability to choose the right steps will greatly determine the player wins. However, to be able to choose the exact steps required a decision tree (spanning tree) which would also require a large storage, but flexible. Here will be proven whether the circular linkedlist able to help address the required needs.

In an prototype based on the Windows operating system that was built using the Java programming language, the application is expected to provide more benefits in terms of flexibility of use of storage.

**Keyword :** Linked list, congklak



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Circular Linked list (sebuah model dari Linked List) merupakan sebuah konsep pengalokasian memori yang ditawarkan pada beberapa bahasa pemrograman seperti Pascal, C++, juga Java untuk menangani penyimpanan data pada memori secara dinamis. Dengan konsep penyimpanan data yang dinamis, Circular Linked List menawarkan solusi pemecahan masalah penyimpanan data yang tidak dimiliki oleh larik.

Linked List dalam bentuk yang lebih luas sering digunakan oleh banyak programmer untuk melakukan penyimpanan data-data yang berorientasi pada record seperti data pegawai, data buku dan lain sebagainya. Dalam kasus yang lain linked list juga sering dimanfaatkan untuk pemecahan kasus-kasus yang membutuhkan media penyimpanan dinamis non record seperti penyimpanan spanning tree, sparse matriks, stack, queue, dan lain sebagainya. Hal ini sedikit banyak disebabkan karena linked list menawarkan kemudahan dalam pengalokasian memori (heap) yang dapat diinisiasi pada saat diperlukan, termasuk pendalokasian memori pada saat memori telah selesai digunakan. Kemudahan lain yang ditawarkan adalah fleksibilitasnya dalam membuat link (hubungan) dari sebuah node ke node yang lain yang diinginkan.

Penelitian ini akan membahas penerapan circular linked list dalam pengelolaan sebuah memori dinamis berupa spanning tree dalam sebuah proses pencarian keputusan dari sebuah game congklak secara heuristik. Game congklak adalah game yang mengharuskan kedua pemainnya memilih satu dari sekian alternatif langkah yang mungkin dalam usahanya meraih kemenangan. Alternatif-



alternatif langkah yang mungkin tersebut kemudian dijabarkan dalam sebuah spanning tree yang nantinya akan dioptimasi menggunakan teknik Depth Limited Search (DLS) untuk memperoleh langkah dengan peluang kemenangan tertinggi. Selain itu circular linked list juga akan diterapkan dalam pengelolaan variabel-variabel penyimpan biji permainan yang mengharuskan adanya ketersambungan antara lubang-lubang yang ada di dalamnya.

## **1.2 Rumusan Permasalahan**

Bagaimana implementasi circular linked list pada aplikasi game congklak serta bagaimana diperoleh alternatif langkah yang mungkin tempuh pemain untuk menghasilkan keuntungan permainan yang optimal.

## **1.3 Batasan Masalah**

Batasan masalah pada penelitian ini meliputi :

- A. Aplikasi game yang dibuat untuk dimainkan oleh 2 orang
- B. Pada permainan disediakan fasilitas “Bantuan Analisa” untuk Bermain.  
Fasilitas Bantuan Analisa dibangun dengan algoritma DLS.
- C. Penerapan Circular linked list dilakukan dalam 2 hal : (1) pada pembuatan struktur data permainan dan (2) pengelolaan spanning tree hasil dari operasi algoritma DLS.

## **1.4 Tujuan Penelitian**

Merekayasa sebuah game berupa permainan tradisional congklak dengan menerapkan konsep circular linked list.



### **1.5 Manfaat Penelitian**

Memberikan alternatif cara bermain congklak

### **1.6 Metodologi Penelitian**

Metodologi yang digunakan pada penelitian ini dijabarkan dalam urutan langkah-langkah sebagai berikut :

- A. Penentuan Alat dan Bahan
- B. Perancangan Struktur Data
- C. Perancangan Flowchart dan Algoritma
- D. Perancangan Fungsi-fungsi pendukung
- E. Pengujian Aplikasi

### **1.7 Sistematika Penulisan**

Sistematika penulisan penelitian ini diuraikan dengan urutan penulisan sebagai berikut :

- A. Bab I Pendahuluan meliputi latar belakang, perumusan dan batasan permasalahan, tujuan dan manfaat penelitian, metodologi penelitian, dan sistematika penulisan.
- B. Bab II Tinjauan Pustaka
- C. Bab III Landasan Teori
- D. Bab IV Analisis dan Rancangan Sistem
- E. Bab V Implementasi dan Pembahasan
- F. Bab VI Kesimpulan
- G. Daftar Pustaka
- H. Lampiran



## BAB II

### TINJAUAN PUSTAKA

Penelitian mengenai pembuatan permainan congklak digital sekalipun belum banyak dijumpai namun pernah dilakukan pada penelitian pembuatan aplikasi congklak berbasis *bluetooth*. (Philipus, 2011). Pada aplikasi tersebut permainan congklak dimainkan oleh 2 orang pemain menggunakan handphone. Penekanan penelitian tersebut terletak pada aspek komunikasi perangkat keras di mana kedua pemain masing-masing bermain congklak menggunakan handphone yang dilengkapi fasilitas *bluetooth*. Kedua pemain yang telah terhubung secara jaringan dapat melakukan permainan dengan posisi saling berlawanan satu sama lain. Dalam penelitian di atas permainan congklak digital yang telah dibangun belum memiliki unsur cerdas yang memungkinkan komputer dapat bermain melawan manusia, sehingga untuk mengoperasikannya membutuhkan paling tidak 2 orang pemain dengan mengambil posisi saling berhadapan (berlawanan). Permainan ini juga belum dapat mengakomodasi kepentingan pemain untuk menganalisa langkah-langkah yang akan dia tempuh untuk memperoleh kemenangan sehingga model permainan ini perlu dikembangkan dengan memasukkan unsur cerdas di dalamnya.

Penelitian mengenai implementasi konsep circular linked list pernah dilakukan di Universitas Kristen Duta Wacana (Agung, 1996). Pada penelitian tersebut circular linked list diterapkan dalam pembuatan visualisasi eliminasi gauss pada sparse matrix. Circular linked list yang diterapkan dalam pembuatan struktur sparse matrix memberikan efisiensi penggunaan memori untuk penyimpanan dengan tingkat fill in 30 persen.



## BAB III

### LANDASAN TEORI

#### 3.1 Permainan Congklak

##### 3.1.1. Definisi

Congklak adalah salah satu jenis permainan tradisional yang cukup dikenal di Indonesia, namun demikian di berbagai daerah orang mengenal permainan ini dengan nama yang berbeda. Di beberapa daerah di Sumatra dan Melayu, orang menyebut permainan ini dengan istilah *congklak*, sementara di Jawa, permainan ini lebih dikenal dengan nama *dakon*, *dhakon* atau *dhakonan*. Di Lampung permainan ini lebih dikenal dengan nama *dentuman lamban* sedangkan di Sulawesi permainan ini lebih dikenal dengan nama *Mokaotan*, *Maggalecteng*, *Aggalacang* dan *Nogarata*. Dalam bahasa Inggris, permainan ini disebut dengan nama *Mancala*.



**Gambar 3.1 Bentuk Fisik Congklak yang terbuat dari kayu**

Alat yang digunakan dalam permainan ini terdiri dari 2 jenis yaitu papan congklak dan biji congklak. Pada jaman dahulu papan congklak dibuat orang dengan menggunakan bahan kayu sedangkan bijinya dapat menggunakan cangkang kerang, biji-biji, atau batu-batuan.



Orang memainkan permainan congklak ini dengan cara memindah-mindahkan biji congklak diatas papan congklak, sedemikian rupa sehingga dihasilkan **total poin** berupa biji terbanyak yang mampu dikumpulkan oleh pemain. Permainan ini dilakukan oleh dua orang pemain dengan saling berhadapan, dan saling mengalahkan. Dalam permainan ini pemain dituntut untuk dapat berpikir analitis sehingga mampu menentukan satu dari banyak lubang untuk **dijalankan** sedemikian sehingga diperoleh keuntungan (total poin) sebanyak-banyaknya atas lawan.

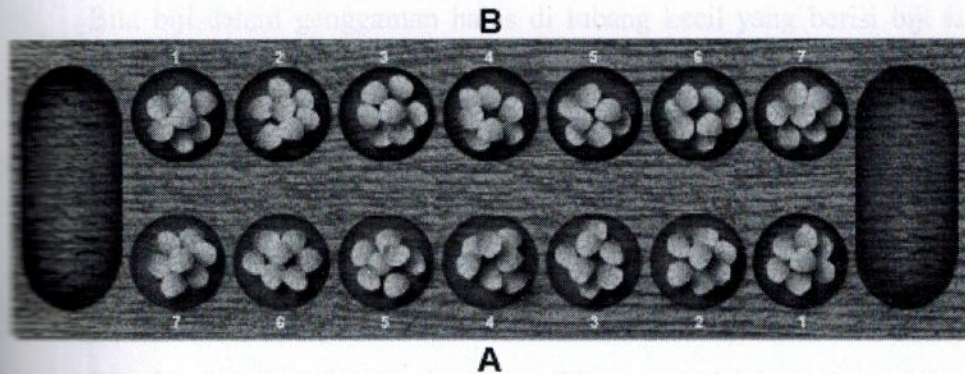


**Gambar 3.2 Permainan Congklak yang sedang dimainkan**

### **3.1.2 Bentuk Papan Congklak**

Papan congklak terdiri dari banyak lubang yang dibuat sedemikian rupa sehingga masing-masing lubang dapat menampung sejumlah biji congklak. Ada 16 lubang pada papan congklak yang terdiri atas 14 **lubang kecil** yang saling berhadapan, dan 2 **lubang besar** (disebut **lumbung**) pada kedua sisinya. Setiap 7 lubang kecil pada sisi pemain dan lumbung yang berada di sebelah kirinya dianggap sebagai wilayah pemain (**wilayah pemain**), sedangkan yang berada di sisi sebaliknya adalah milik lawan.





**Gambar 3.3 Congklak dan Bijinya**

### 3.1.3. Cara Memainkan Congklak

Pada awal permainan pada keenambelas *lubang kecil* masing-masing diisi dengan biji sebanyak 7 buah, sedangkan pada kedua *lambung* dibiarkan dalam keadaan kosong. Secara keseluruhan banyaknya biji di masing-masing sisi pemain (A maupun B) adalah sebanyak 49 buah, serta total keseluruhan biji yang dibutuhkan untuk permainan baik A dan B adalah 98 buah.

Untuk memainkan permainan congklak, kedua pemain harus terlebih dahulu menentukan salah satu diantara kedua pemain tersebut yang berhak mendapat *giliran* pertama untuk bermain. Pemain yang mendapat *giliran* untuk bermain dapat memilih salah satu ketujuh lubang kecil yang ada di *wilayah pemain* kemudian memainkannya dengan *1 genggam* yaitu mengambil seluruh biji yang ada di dalam lubang tersebut, kemudian meletakkan satu per satu (*1 langkah* demi *1 langkah*) biji-biji tersebut ke dalam lubang-lubang ada di sebelahnya baik *lubang kecil* maupun *lambung* sesuai dengan *rute permainan* (kecuali *lambung* milik lawan).



Bila biji dalam genggaman habis di lubang kecil yang berisi biji lain (bukan lubang kosong dan bukan lumbung), maka pemain dapat mengambil semua biji yang ada pada lubang tersebut dan melanjutkan permainan sebanyak 1 genggaman lagi.

Bila biji habis di ***lumbung pemain***, maka dia dapat memilih lagi salah satu lubang kecil yang ada di ***wilayah pemain*** untuk "***dimainkan***" (jika ada).

Bila biji habis di lubang kosong miliknya, maka ia berhenti bermain (***mati langkah***) tapi boleh mengambil seluruh biji ***jarahan*** yang ada di sisi lawan yang posisinya tepat berada di seberangannya. Tetapi bila pemain ***mati langkah*** di lubang kosong milik lawan maka ia berhenti bermain dan tidak mendapatkan apa-apa.

Apabila pemain berhenti (***mati langkah***), maka hak untuk bermain akan diberikan kepada lawannya. Pemain juga akan kehilangan hak bermain jika pemain tidak memiliki lubang untuk dimainkan.

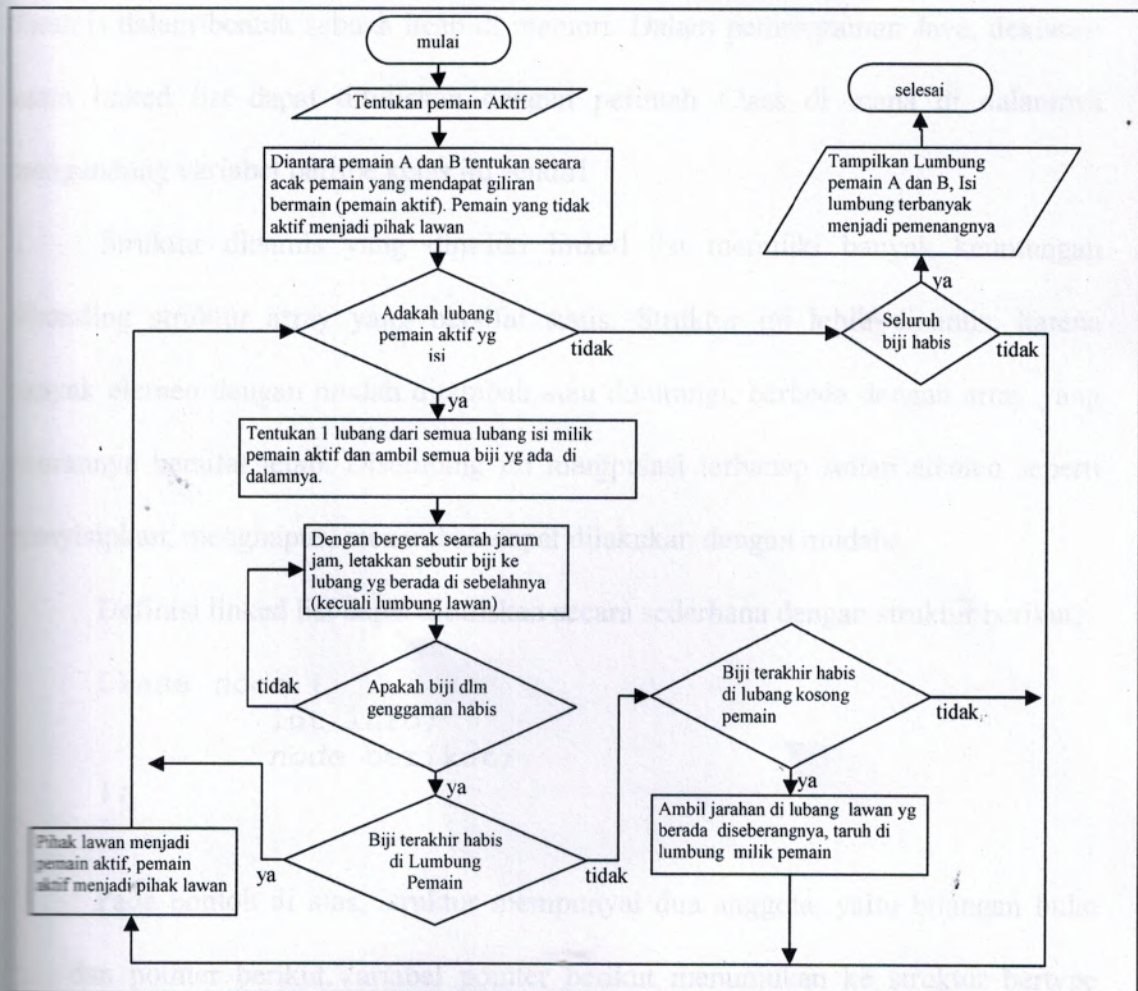
Permainan dianggap selesai bila sudah tidak ada biji lagi yang dapat diambil pada lubang kecil baik milik pemain maupun milik lawan (seluruh biji sudah berada di lumbung kedua pemain).

Dalam menentukan pemenang, pemain yang pada akhir permainan berhasil memperoleh biji terbanyak (total poin terbanyak) di dalam lumbungnya akan keluar sebagai pemenang.

### 3.1.4. Flowchart Permainan Congklak

*Rule of game* permainan congklak tersebut dipaparkan dalam flowchart sebagaimana tersaji dalam Gambar 3.4.





**Gambar 3.4 Flowchart Rule of Game Permainan Congklak**

### 3.1.5. Permainan Congklak Digital

Permainan Congklak digital adalah permainan congklak yang dilakukan secara digital menggunakan perangkat komputer . Permainan congklak digital ditampilkan secara visual melalui layar komputer. Interaksi antara pemain dan congklak dilakukan dengan menggunakan keyboard dan mouse.

### 3.2 Circular Linked List

Circular Linked List adalah sebuah model dari Linked List, yang terdiri dari node yang disusun sedemikian rupa sehingga saling berkaitan satu sama lain membentuk sebuah untaian data. Node-node dalam linked list dibangun secara



dinamis dalam bentuk sebuah heap di memori. Dalam pemrograman Java, deklarasi suatu linked list dapat dituliskan dengan perintah Class di mana di dalamnya mengandung variabel bertipe kelas itu sendiri.

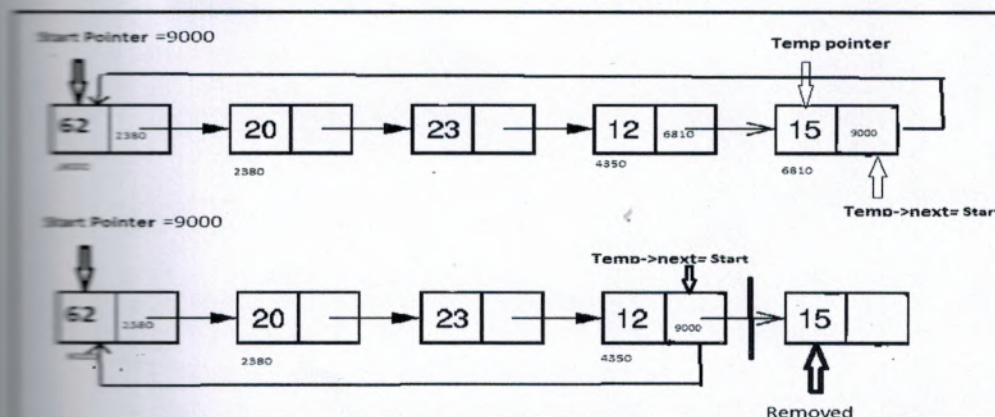
Struktur dinamis yang dimiliki linked list memiliki banyak keuntungan dibanding struktur array yang bersifat statis. Struktur ini lebih dinamis, karena banyak elemen dengan mudah ditambah atau dikurangi, berbeda dengan array yang ukurannya bersifat tetap. Disamping itu manipulasi terhadap setiap elemen seperti menyisipkan, menghapus, menambah dapat dilakukan dengan mudah.

Definisi linked list dapat dituliskan secara sederhana dengan struktur berikut;

```
Class node {
    int info;
    node berikut;
};
```

Gambar 3.6 Circular Linked List pada Tree

Pada contoh di atas, struktur mempunyai dua anggota, yaitu bilangan bulat *info* dan pointer berikut. Variabel pointer berikut menunjukan ke struktur bertipe *node*, yang sama dengan deklarasi kelas induknya.

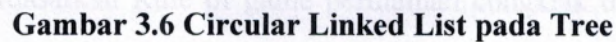


Gambar 3.5 Circular Linked list



```

graph TD
    F((F)) --> B((B))
    F((F)) --> G((G))
    B((B)) --> A((A))
    B((B)) --> D((D))
    G((G)) --> I((I))
    G((G)) --> H((H))
    D((D)) --> C((C))
    D((D)) --> E((E))
    A((A)) -.-> F((F))
    D((D)) -.-> F((F))
    I((I)) -.-> F((F))
    H((H)) -.-> F((F))
    C((C)) -.-> F((F))
    E((E)) -.-> F((F))
  
```





## **BAB IV**

### **ANALISA DAN PERANCANGAN**

Penelitian ini menggunakan metode pengembangan perangkat lunak yang terdiri dari tahap-tahap sebagai berikut :

#### **4.1 Spesifikasi Kebutuhan**

Perlengkapan teknis yang digunakan dalam penelitian ini adalah seperangkat komputer dengan spesifikasi prosesor Core Duo, Memori 1 GB, Harddisk 160 GB dengan sistem operasi Windows XP. Pembuatan aplikasi dilakukan menggunakan tools Netbeans IDE 6.7.1 dengan bahasa Java..

#### **4.2 Tahap Analisis dan Perancangan Sistem**

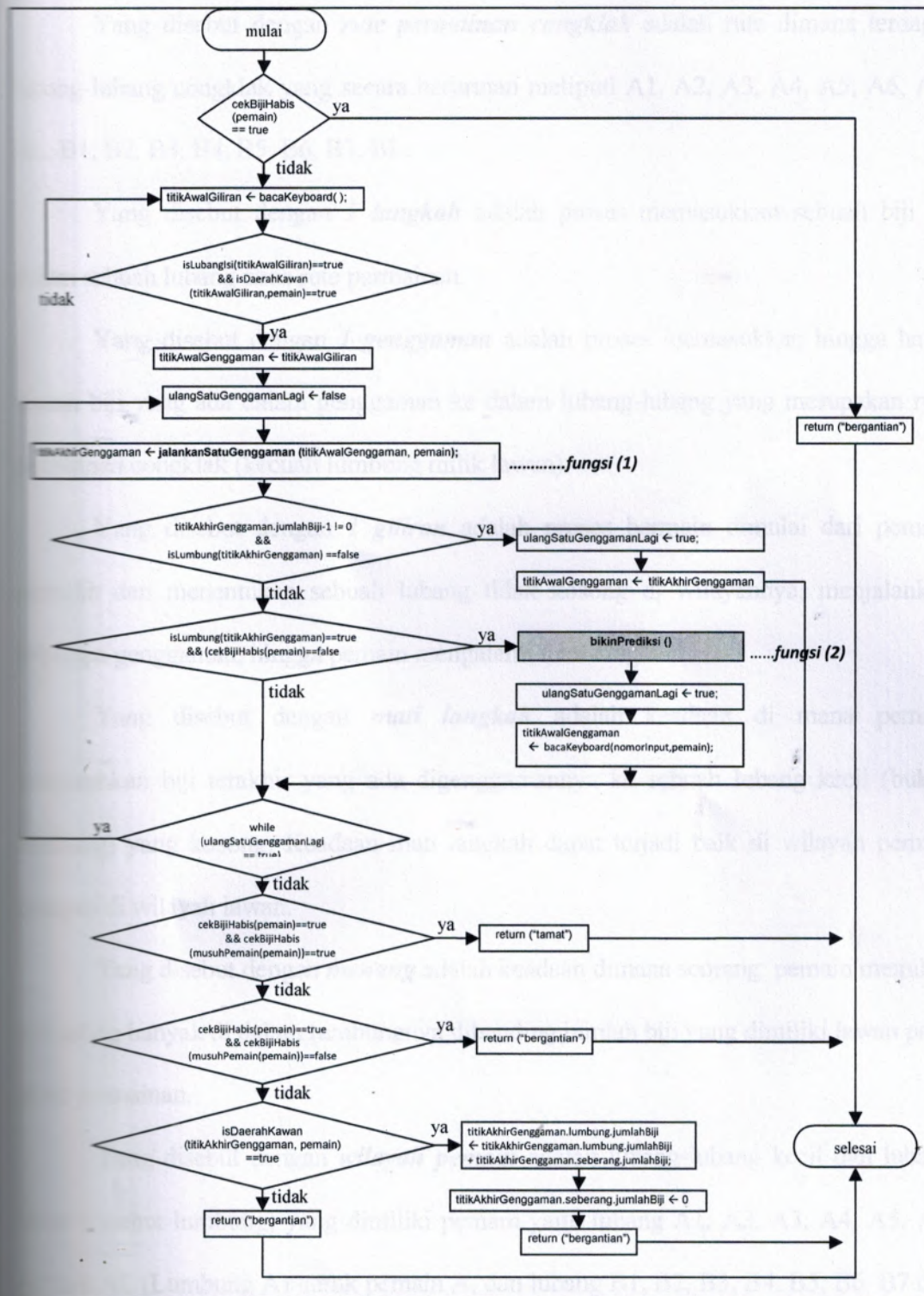
Pekerjaan dalam penelitian ini adalah dengan merencanakan sebuah program berbahasa java berdasarkan Rule of game permainan congklak dengan unsur-unsur meliputi struktur data, circular linked list, algoritma DLS dan fungsi-fungsi lain.

#### **Perancangan Algoritma dan Program Java**

Tahap awal dari pembuatan permainan congklak digital ini adalah membuat algoritma untuk menterjemahkan *rule of game* ke dalam langkah-langkah program dalam bahasa pemrograman java. Adapun “proses bermain” dari game congklak di sini dilakukan dengan mengubah nilai-nilai dari setiap lubang dan lubang, sedemikian rupa mengikuti aturan permainan .

Flowchart pada gambar 4.1 menjelaskan bagaimana “proses bermain”nya seorang pemain congklak dimulai dari pemain mendapat giliran untuk bermain, pemain mengelola permainan sebaik mungkin, sampai dengan pemain tersebut kehilangan haknya untuk bermain dan memberikan giliran permainan kepada lawan.





**Gambar 4.1 Flowchart dari fungsi untuk menjalankan 1 giliran**

Penamaan variabel direpresentasikan menurut istilah-istilah sebagai berikut.



Yang disebut dengan *rute permainan congklak* adalah rute dimana terdapat lubang-lubang congklak yang secara berurutan meliputi A1, A2, A3, A4, A5, A6, A7, AL, B1, B2, B3, B4, B5, B6, B7, BL.

Yang disebut dengan *1 langkah* adalah proses memasukkan sebuah biji ke dalam sebuah lubang pada rute permainan.

Yang disebut dengan *1 genggaman* adalah proses memasukkan hingga habis semua biji yang ada dalam genggaman ke dalam lubang-lubang yang merupakan rute permainan congklak (kecuali lumbung milik lawan).

Yang disebut dengan *1 giliran* adalah proses bermain dimulai dari pemain memilih dan menentukan sebuah lubang tidak kosong di wilayahnya, menjalankan beberapa genggaman, hingga pemain mengalami mati langkah.

Yang disebut dengan *mati langkah* adalah keadaan di mana pemain menjatuhkan biji terakhir yang ada digenggamannya ke sebuah lubang kecil (bukan lumbung) yang kosong. Keadaan mati langkah dapat terjadi baik di wilayah pemain maupun di wilayah lawan.

Yang disebut dengan *menang* adalah keadaan dimana seorang pemain memiliki biji paling banyak di dalam lumbungnya dibanding jumlah biji yang dimiliki lawan pada akhir permainan.

Yang disebut dengan *wilayah pemain* adalah lubang-lubang kecil dan lubang besar (disebut lumbung) yang dimiliki pemain yaitu lubang A1, A2, A3, A4, A5, A6, A7 dan AL (Lumbung A) untuk pemain A, dan lubang B1, B2, B3, B4, B5, B6, B7 dan BL (Lumbung B) untuk pemain B.

Yang disebut dengan *wilayah lawan* adalah lubang-lubang kecil dan lubang besar (disebut lumbung) yang bukan termasuk wilayah pemain.



Yang disebut dengan *lubang kecil* adalah A1...A7 dan B1...B7

Yang disebut dengan *lumbung* adalah LA dan LB

Yang disebut dengan *lumbung pemain* adalah lubang besar yang dimiliki pemain yaitu lubang LA untuk pemain A, dan lubang LB untuk pemain B.

Yang disebut dengan *jarahan* adalah semua biji yang ada dalam lubang milik lawan yang letaknya tepat berada disebelang lubang milik pemain dimana pemain mengalami mati langkah (yaitu mati langkah di wilayah sendiri).

Yang disebut dengan *biji habis* adalah habisnya biji dari genggamannya setelah pemain menjalankan 1 genggamannya.

Yang disebut dengan *mati langkah* adalah apabila biji terakhir habis pada lubang kosong hingga pemain tidak dapat melanjutkan langkah.

Berdasarkan flowchart pada gambar 4.1 kemudian dapat ditulis script program sebagaimana Program 4.1.

#### Program 4.1

```
public static String jalankanSatuGiliran(char pemain)
{ String nilaiReturn = "";
  if (cekBijiHabis(pemain)==true)
    //jika semua lubang pemain kosong maka tidak dapat jalan,
    //giliran diserahkan ke musuh
    { nilaiReturn = "GANTIAN";
    }
  else
  { int nomorInput=1;
    bikinPrediksi(pemain,a1.jumlahBiji,a2.jumlahBiji,a3.jumlahBiji,
      a4.jumlahBiji
      ,a5.jumlahBiji,a6.jumlahBiji,a7.jumlahBiji,a8.jumlahBiji
      ,b1.jumlahBiji,b2.jumlahBiji,b3.jumlahBiji,b4.jumlahBiji
      ,b5.jumlahBiji,b6.jumlahBiji,b7.jumlahBiji,b8.jumlahBiji);
    titikAwalGiliran = bacaKeyboard(nomorInput,pemain);
    nomorInput++;

    if((isLubangIsi(titikAwalGiliran)==true)
      &&(isDaerahKawan(titikAwalGiliran, pemain)==true))
      //jika titikAwalGiliran sah
      //yaitu Lubang tidak kosong & wilayah pemain
      { simpulCongklak titikAwalGenggamannya;
```



```

simpulCongklak titikAkhirGenggaman;
boolean ulangSatuGenggamanLagi;

int nomorGenggaman = 1;
titikAwalGenggaman = titikAwalGiliran;

do
{
    ulangSatuGenggamanLagi = false;
    titikAkhirGenggaman
        = jalankanSatuGenggaman(titikAwalGenggaman, pemain);
    if(((titikAkhirGenggaman.jumlahBiji-1)!=0)
        &&(isLambung(titikAkhirGenggaman)==false))
        // jika biji jatuh pada lubang yg tidak kosong,
        // dan bukan lambung --> LANJUT
    {
        ulangSatuGenggamanLagi = true;
        nomorGenggaman++;
        titikAwalGenggaman = titikAkhirGenggaman;
    }
    if ((isLambung(titikAkhirGenggaman)==true)
        && (cekBijiHabis(pemain)==false))
        //jika biji jatuh pada lambung & biji pemain belum habis
        // maka BACA KEYBOARD, LANJUT
    {
        bikinPrediksi(pemain
            ,a1.jumlahBiji,a2.jumlahBiji,a3.jumlahBiji,a4.jumlahBiji
            ,a5.jumlahBiji,a6.jumlahBiji,a7.jumlahBiji,a8.jumlahBiji
            ,b1.jumlahBiji,b2.jumlahBiji,b3.jumlahBiji,b4.jumlahBiji
            ,b5.jumlahBiji,b6.jumlahBiji,b7.jumlahBiji,b8.jumlahBiji);

        ulangSatuGenggamanLagi = true;
        titikAwalGenggaman = bacaKeyboard(nomorInput,pemain);
        nomorInput++;
        nomorGenggaman++;
    }
} while (ulangSatuGenggamanLagi == true);

//MENGANALISA TITIK DIMANA BIJI TERAKHIR JATUH
if((cekBijiHabis(pemain)==true)
    && (cekBijiHabis(musuhPemain(pemain))==true))
    //jika semua biji pemain habis, dan semua biji musuh pemain
    // juga habis, permainan berakhir
    { nilaiReturn = "TAMAT";
    }
else if ((cekBijiHabis(pemain)==true)
    && (cekBijiHabis(musuhPemain(pemain))==false))
    //jika semua biji pemain habis, dan biji musuh masih punya,
    // permainan berganti
    { nilaiReturn = "GANTIAN";
    }
else if (isDaerahKawan(titikAkhirGenggaman, pemain)==true)
    //jika biji habis dilubang kosong pemain & bukan lambung,
    // giliran selesai, berganti musuh
    { titikAkhirGenggaman.lambung.jumlahBiji =
        titikAkhirGenggaman.lambung.jumlahBiji
        + titikAkhirGenggaman.seberang.jumlahBiji;
        titikAkhirGenggaman.seberang.jumlahBiji = 0;
        //kosongkan lubang lawan
        nilaiReturn = "GANTIAN";
    }
}

```

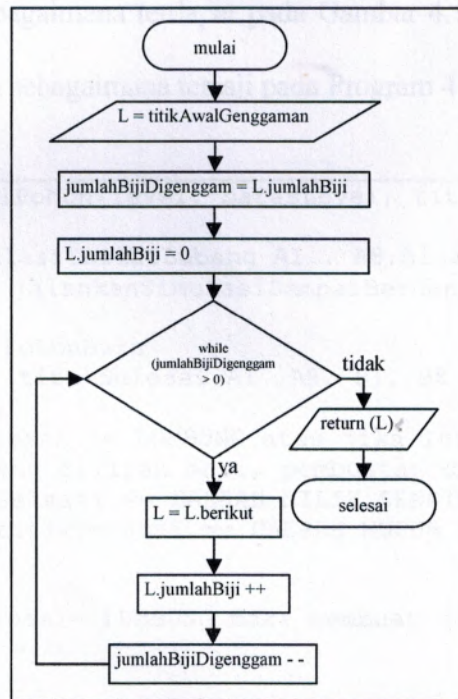


```

else //jika biji habis dikandang lawan,
    //giliran selesai,.berganti ke musuh
{ System.out.println("Biji habis didaerah LAWAN,
    anda tidak mendapat apa-apa");
    System.out.println("SEKARANG GILIRAN MUSUH ANDA..");
    nilaiReturn = "GANTIAN";
}
}
}
return(nilaiReturn);
}

```

Fungsi (1) sebagaimana terdapat pada Gambar 4.1 dapat diurai lebih detail lagi ke dalam sub flowchart sebagaimana tersaji pada Gambar 4.2. Sub flowchart tersebut menjelaskan bagaimana perubahan nilai-nilai jumlah biji pada setiap lubang dilakukan, dimulai dari diambilnya seluruh biji pada sebuah lubang yang dipilih pemain, mendistribusikan satu-persatu biji ke lubang di sebelahnya, hingga habisnya biji yang ada dalam genggam.



**Gambar 4.2 Flowchart untuk menjalankan 1 genggam**



Berdasarkan flowchart pada gambar 4.2 kemudian dapat ditulis subrutin program sebagaimana Program 4.2. Subrutin ini adalah untuk menjalankan 1 genggaman.

### Program 4.2

```
public static simpulCongklak jalankanSatuGenggaman(simpulCongklak
titikAwalGenggaman, char pemain)
{ simpulCongklak L;
  int jumlahBijiDigenggam = 0;
  L = titikAwalGenggaman;      //menentukan lubang yang hendak diambil bijinya
  jumlahBijiDigenggam = L.jumlahBiji; //mengambil biji dan menggenggamnya
  L.jumlahBiji= 0; //mengosongkan lubang yg tlah diambil bijinya mjd kosong
  while(jumlahBijiDigenggam > 0) //ulangi hingga biji dlm genggaman habis
  { L = L.berikut;
    if ( (pemain=='a') && (L.namaLubang!="LB8")
        || (pemain=='b') && (L.namaLubang!="LA8"))
      //jika pemain=a, telusuri semua lubang KECUALI LUMBUNG B
      //jika pemain=b, telusuri semua lubang KECUALI LUMBUNG A
      { L.jumlahBiji++;
        jumlahBijiDigenggam--;
      }
  }
  return(L);
}
```

Fungsi (2) sebagaimana terdapat pada Gambar 4.1 dapat diuraikan lebih detail lagi ke dalam subrutin sebagaimana tersaji pada Program 4.3.

### Program 4.3

```
fungsi bikinSimpulPohon(level, batasLevel, titikMulai, pemain, A1..
A8, B1.. B8 )
{ setCongklakSimulasi(nilaiLubang A1.. A8, nilaiLubang B1.. B8);
  titikSelesai ← jalankanSimulasiSampaiBerhenti(titikMulai,
pemain);
  ciptakan simpulPohonBaru
  isi titikMulai, titikSelesai, A1..A8, B1..B8 dgn kondisi lubang terkini
  jika pemain = A
    jika titikSelesai != LUMBUNG atau jika level > 10
      maka giliran mati, pembuatan cabang berhenti
    { jika titikSelesai == DAERAH MILIK SENDIRI --> dapat jarahan
      else jika titikSelesai == DAERAH MUSUH --> tidak mendapat
jarahan
    }
    else titikSelesai= LUMBUNG maka membuat cabang baru secara
rekursif
    {
      jika lubang A1 tidak kosong, dan level < batas level
      { bikinSimpulPohon(level+1, batasLevel, cabang1, pemain,
A1..A8, B1..B8)
        setCongklakSimulasi(nilaiLubang A1.. A8,
          nilaiLubang B1.. B8);
      }
```



```

    }
    jika lubang A2 tidak kosong, dan level < batas level
    { bikinSimpulPohon(level+1, batasLevel, cabang1,
        pemain, A1..A8, B1..B8)
        setCongklakSimulasi(nilaiLubang A1.. A8,
            nilaiLubang B1.. B8);
    }
    ...
    ...
    dst sama untuk lubang A1 hingga A8
}
}
else... jika pemain = B lakukan hal yang sama dengan pemain=A
return(simpulPohonBaru);
}

```

### Perancangan Struktur Data

Pada bagian ini dilakukan perancangan struktur data yang akan digunakan untuk menyimpan nilai-nilai variabel yang mewakili semua lubang pada permainan congklak.

Media penyimpan yang digunakan dalam pembuatan aplikasi pada penelitian ini adalah linkedlist. Struktur obyek dan kelas dibangun dan definisikan sebagaimana terlihat pada Program 4.4.

#### Program 4.4

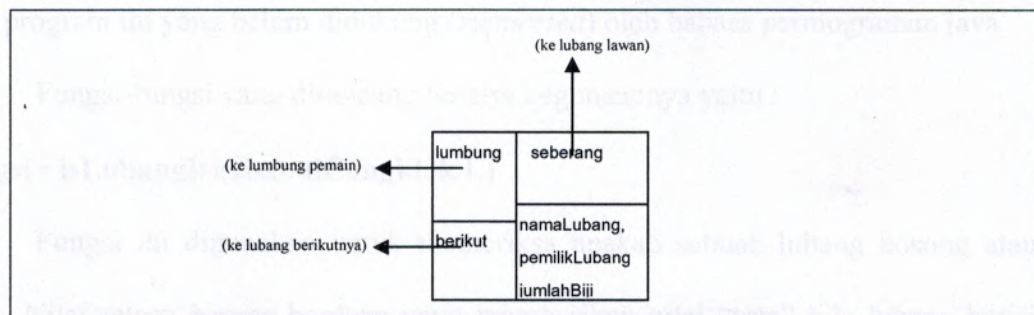
```

class simpulCongklak
{
    public String      namaLubang;
    public char pemilikLubang;
    public int jumlahBiiji;
    public simpulCongklak   berikut;
    public simpulCongklak   seberang;
    public simpulCongklak   lambung;
}

```

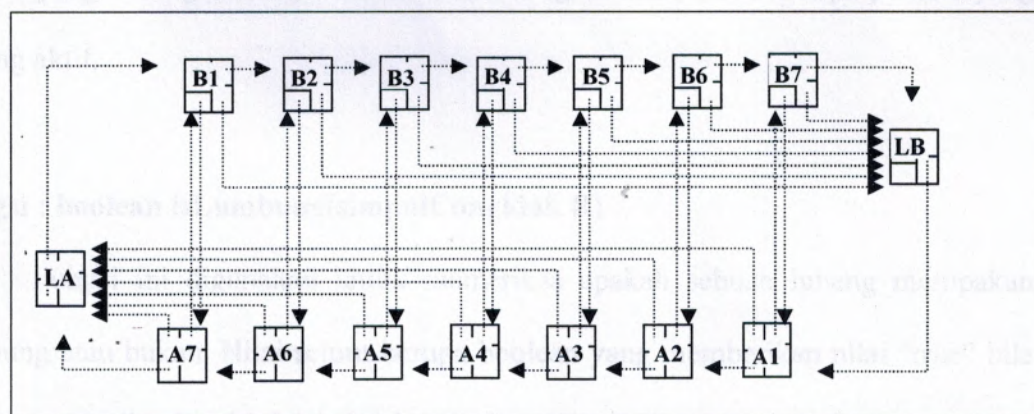


Dari deklarasi struktur kelas di atas, dapat dilihat skema simpul yang terbentuk dapat sebagaimana Gambar 4.3.



**Gambar 4.3 Skema simpul yang digunakan**

Setiap simpul yang tercipta sebagaimana digambarkan pada Gambar 4.3 diatur untuk selalu menunjuk ke simpul di sebelahnyanya sedemikian rupa sehingga membentuk rangkaian simpul yang saling berhubungan sesuai *rule permainan*. *Pointer* [namaLubang].berikut diatur untuk menunjuk ke simpul lubang sebelah kirinya (searah jarum jam) sedangkan pointer [namaLubang].seberang diatur menunjuk ke simpul yang berada di seberangnya. Pointer [namaLubang].lambung dibuat menunjuk ke simpul lubang milik pemain sedemikian sehingga membentuk jaringan simpul (linked list) sebagaimana disajikan dalam Gambar 4.4.



**Gambar 4.4 Skema simpul (linked list) pada aplikasi congklak**



## **Perancangan Fungsi-fungsi**

Pada bagian ini dilakukan perancangan fungsi-fungsi yang akan digunakan oleh program ini yang belum didukung (*supported*) oleh bahasa pemrograman java.

Fungsi-fungsi yang dirancang beserta kegunaannya yaitu :

### **Fungsi : isLubangIsi(simpulCongklak L)**

Fungsi ini digunakan untuk memeriksa apakah sebuah lubang kosong atau tidak. Nilai return berupa boolean yang memberikan nilai "true" bila lubang berisi dan "false" jika lubang kosong. Argumen yang diperlukan untuk menggunakan fungsi ini adalah pointer yang sedang menunjuk ke sebuah lubang tertentu.

### **Fungsi : boolean isDaerahKawan(simpulCongklak R,char pemain)**

Fungsi ini digunakan untuk memeriksa apakah sebuah lubang merupakan milik pemain atau milik lawan. Nilai return berupa boolean yang memberikan nilai "true" bila lubang yang bersangkutan adalah milik pemain, dan "false" jika lubang adalah milik lawan. Argumen yang diperlukan untuk menggunakan fungsi ini adalah pointer yang sedang menunjuk ke sebuah lubang tertentu, beserta siapa pemain yang sedang aktif.

### **Fungsi : boolean isLambung(simpulCongklak R)**

Fungsi ini digunakan untuk memeriksa apakah sebuah lubang merupakan lambung atau bukan. Nilai return berupa boolean yang memberikan nilai "true" bila lubang yang dimaksud adalah lambung, dan "false" jika lubang bukan merupakan lambung. Argumen yang diperlukan untuk menggunakan fungsi ini adalah pointer yang sedang menunjuk ke sebuah lubang tertentu.



**Fungsi : boolean cekBijiHabis(char pemain)**

Fungsi ini digunakan untuk memeriksa apakah masih ada diantara lubang milik pemain yang masih berisi biji.. Nilai return berupa boolean yang memberikan nilai “true” bila semua lubang milik pemain sudah kosong, dan “false” jika masih ada lubang milik pemain yang berisi biji. Argumen yang diperlukan untuk menggunakan fungsi ini adalah siapa pemain yang sedang aktif.

**menjalankan 1 genggaman()**

Fungsi ini digunakan untuk mengeksekusi permainan sebanyak 1 genggaman, dimulai dari pemain mengambil biji dari sembarang lubang miliknya, mendistribusikan biji-biji tersebut ke lubang-lubang di sebelahnya, hingga biji dalam genggaman habis. Nilai return berupa pointer yang berupa lokasi lubang tempat dimana biji terakhir jatuh. Argumen yang diperlukan untuk menggunakan fungsi ini adalah pointer yang menunjuk ke sebuah lubang yang telah dipilih untuk dimainkan.

**Perancangan Depth Limited Search (DLS)**

Pada bagian ini dilakukan pengimplementasian Algoritma *Depth Limited Search* (DLS) untuk membuat analisis atas alternatif-alternatif langkah yang mungkin ditempuh termasuk penelusuran cabang-cabang kemungkinan yang baru yang ditelusur secara mendalam (namun terbatas) yang berpotensi menghasilkan kemenangan terbesar (optimal).

Implementasi DLS pada program dipaparkan pada algoritma sebagaimana tersaji pada Program 4.5.



#### Program 4.5

```
fungsi bikinSimpulPohon(level, batasLevel, titikMulai, pemain, A1.. A8,
B1.. B8 )
{
    setCongklakSimulasi(nilaiLubang A1.. A8,nilaiLubang B1.. B8);
    titikSelesai <-jalankanSimulasiSampaiBerhenti(titikMulai, pemain);
    ciptakan simpulPohonBaru
    isi titikMulai, titikSelesai,A1..A8, B1..B8 dengan kondisi lubang
    terkini
    jika pemain = A
        jika titikSelesai != LUMBUNG atau jika level > 10 pembuatan cabang
        berhenti
        { jika titikSelesai == DAERAH MILIK SENDIRI --> dapat jarahan
          else jika titikSelesai == DAERAH MUSUH --> tidak mendapat jarahan
        }
    else titikSelesai= LUMBUNG maka membuat cabang baru secara rekursif
    { jika lubang A1 tidak kosong, dan level<batas level
      { bikinSimpulPohon(level+1,batasLevel, cabang1, pemain, A1..A8,
                                                                    B1..B8)
        setCongklakSimulasi(nilaiLubang A1.. A8,nilaiLubang B1.. B8);
      }
      jika lubang A2 tidak kosong, dan level<batas level
      { bikinSimpulPohon(level+1,batasLevel, cabang1, pemain, A1..A8,
                                                                    B1..B8)
        setCongklakSimulasi(nilaiLubang A1.. A8,nilaiLubang B1.. B8);
      }
      ...
      ...
      dst sama untuk lubang A1 hingga A8
    }
    else... jika pemain = B lakukan hal yang sama dengan pemain=A
    return(simpulPohonBaru);
}
```

#### Perancangan Antar Muka

Untuk memfasilitasi user dalam berinteraksi dengan sistem dirancang suatu antar muka pengguna (interface) dengan konsep GUI. Antar muka dibangun dibangun menggunakan swing java dalam wujud sebuah form aplikasi.



## BAB V

### IMPLEMENTASI DAN PEMBAHASAN

#### 5.1 Implementasi

Implementasi dari antar muka permainan congklak dapat dilihat pada gambar 5.1.



**Gambar 5.1 Interface Permainan Congklak Digital**

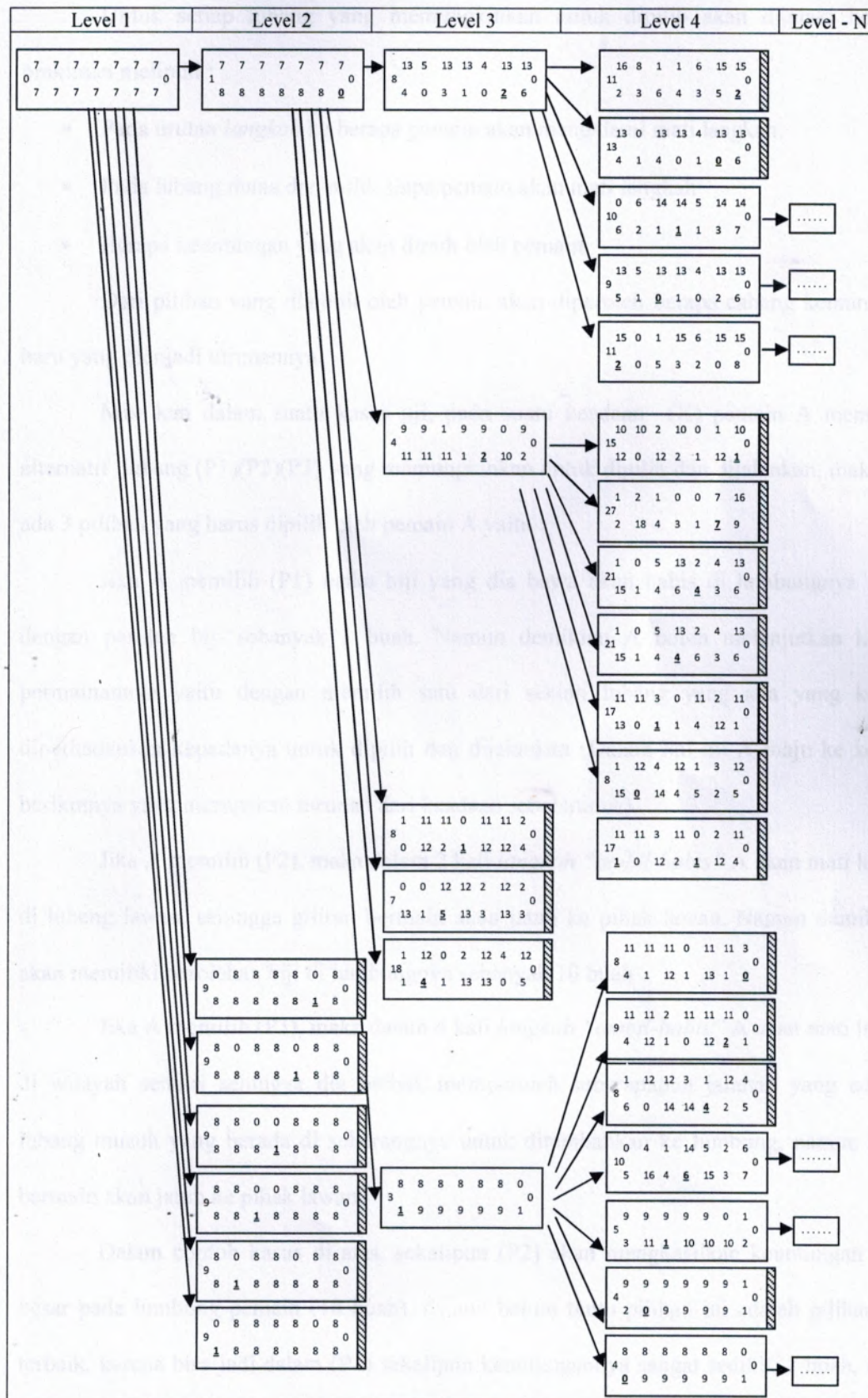
Pemain diharuskan memilih satu diantara tujuh lubang yang tersedia yaitu lubang yang tidak kosong. Setelah lubang dipilih (berwarna kuning) pemain dapat menjalankan simulasinya dengan mengklik tombol “mulai”.

Tombol “Bantuan Analisa” digunakan untuk melihat prakiraan apa yang terjadi atas semua lubang jika lubang dipilih. Prediksi dianalisa berdasarkan situasi terakhir yang ada pada saat tombol tersebut diklik. Prakiraan dihitung berdasarkan analisa DLS pada level kedalaman yang telah ditentukan pada spin edit di bagian bawah form.

#### 5.2 Pembahasan

Pada pengujian aplikasi dilakukan sebuah simulasi permainan yang dimulai dari kondisi awal. Tombol “Bantuan Analisa” memberikan kemungkinan langkah yang mungkin terjadi sebagaimana tersaji pada gambar 5.2.





**Gambar 5.2 Contoh Analisa Kemungkinan pada kedalaman 4 level**



Untuk setiap lubang yang memungkinkan untuk dipilih akan dijajaki beberapa prakiraan meliputi :

- Pada urutan *langkah* ke berapa pemain akan mengalami mati langkah.
- Pada lubang mana dan milik siapa pemain akan mati langkah
- Berapa keuntungan yang akan diraih oleh pemain

Dari pilihan yang diambil oleh pemain akan diperoleh berapa cabang kemungkinan baru yang menjadi turunannya.

Misalkan dalam suatu kasus uji, pada suatu keadaan (X) pemain A memiliki 3 alternatif lubang (P1)(P2)(P3) yang memungkinkan untuk dipilih dan dijalankan, maka akan ada 3 pilihan yang harus dipilih oleh pemain A yaitu :

Jika A memilih (P1) maka biji yang dia bawa akan habis di lumbungnya sendiri dengan peroleh biji sebanyak 1 buah. Namun demikian A boleh melanjutkan kembali permainannya yaitu dengan memilih satu dari sekian lubang yang ada yang kembali diperhadapkan kepadanya untuk dipilih dan dijalankan (Dalam hal ini A maju ke keadaan berikutnya yang merupakan turunan dari keadaan sebelumnya).

Jika A memilih (P2), maka dalam 2 kali *langkah* "*ambil-habis*" A akan mati langkah di lubang lawan, sehingga giliran bermain akan jatuh ke pihak lawan. Namun demikian A akan memiliki perolehan biji di lumbungnya sebanyak 10 buah

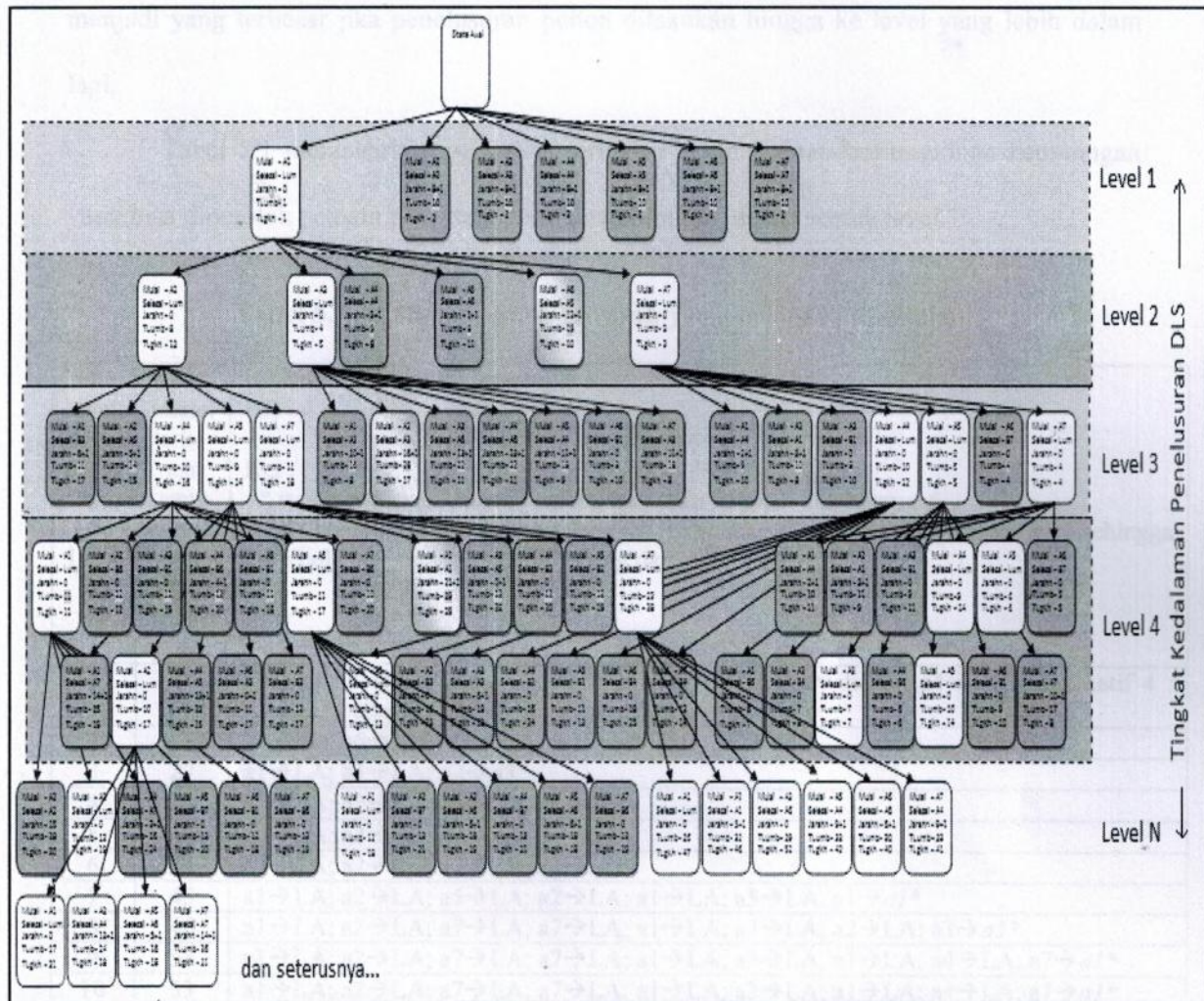
Jika A memilih (P3), maka dalam 4 kali *langkah* "*ambil-habis*" A akan mati langkah di wilayah sendiri sehingga dia berhak memperoleh seberapapun jarahan yang ada dari lubang musuh yang berada di seberangnya untuk ditambahkan ke lumbung, namun giliran bermain akan jatuh ke pihak lawan.

Dalam contoh kasus di atas, sekalipun (P2) akan menghasilkan keuntungan cukup besar pada lumbung pemain (10 buah), namun belum tentu pilihan ini adalah pilihan yang terbaik, karena bisa jadi dalam (P1) sekalipun keuntungannya sangat sedikit, 1 buah, namun



didalamnya terdapat peluang untuk memperoleh keuntungan yang lebih besar lagi pada langkah-langkah turunannya.

Dari kondisi tersebut di atas selanjutnya dapat digambarkan dalam sebuah pohon yang cuplikannya dapat dilihat pada gambar 5.3.



**Gambar 5.3 Pohon Kemungkinan yang dihasilkan**

Root/ akar dari spanning tree diatas menunjukkan state-1 di mana pemain memiliki 7 alternatif pilihan untuk memulai permainan.

Pada setiap level, tampak simpul berwarna putih yang menggambarkan bahwa langkah pemain berakhir di lumbung yang menyebabkan pemain memperoleh hak untuk



melanjutkan kembali permainan. Simpul-simpul putih tersebut selanjutnya menjadi state-2, 3, 4 dan seterusnya.

Nilai keuntungan lubang yang terbesar pada sebuah level atau hingga ke kedalaman level tertentu belum bisa menjadi jaminan bahwa keuntungan itu masih akan tetap menjadi yang terbesar jika penelusuran pohon dilakukan hingga ke level yang lebih dalam lagi.

Tabel 5.1 menjabarkan pengujian terhadap kemungkinan-kemungkinan keuntungan yang bisa diperoleh pemain pada saat permainan dimulai, untuk semua level.

**Tabel 5.1 Analisa langkah-langkah yang mungkin dilakukan**

Jika ditelusur sedalam (N) Level	Keuntungan terbesar yang mungkin diperoleh (K)	Urutan langkah untuk mencapai K (U)  Ket : * = menunjukkan bahwa pemain akan mati langkah di lubang ... * LA menunjukkan bahwa langkah pemain akan jatuh di lubang sehingga diperbolehkan untuk memilih lubang dan main kembali.
1	9	alternatif 1 : $a2 \rightarrow a2^*$ , alternatif 2 : $a3 \rightarrow a3^*$ , alternatif 3 : $a4 \rightarrow a4^*$ , alternatif 4 : $a5 \rightarrow a5^*$ , alternatif 5 : $a6 \rightarrow a6^*$ , alternatif 6 : $a7 \rightarrow a7^*$
2	18	$a1 \rightarrow LA$ ; $a6 \rightarrow a5^*$
3	27	$a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a2 \rightarrow a3^*$
4	38	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow a1^*$
5	38	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow a1^*$
6	38	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow a1^*$
7	45	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a1 \rightarrow a1^*$
8	46	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a3 \rightarrow a5^*$
9	53	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a7 \rightarrow a1^*$
10	53	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a7 \rightarrow a1^*$
11	59	$a1 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a3 \rightarrow LA$ ;
12	63	$a1 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a6 \rightarrow a6^*$
13	69	$a1 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow a2^*$
14	69	$a1 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow a2^*$
15	75	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a4 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow a5^*$
16	76	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow a1^*$
17	77	$a1 \rightarrow LA$ ; $a2 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a3 \rightarrow LA$ ; $a1 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a6 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow LA$ ; $a5 \rightarrow LA$ ; $a7 \rightarrow a1^*$



Analisa terhadap langkah-langkah dapat dilakukan, baik untuk jangkauan kedalaman tertentu bahkan hingga ke dasar, namun demikian DLS memberi batasan penelusuran hingga level tertentu yaitu level yang cukup untuk menghasilkan sebuah kemenangan.

Dalam permainan congklak, syarat cukup untuk dapat menghasilkan kemenangan dapat dihitung dengan rumus :

$$\frac{\text{banyak biji yg digunakan dalam permainan} + 1}{2}$$

sehingga untuk dapat memenangkan permainan, pemain diharuskan memperoleh biji minimal sebanyak :

$$\begin{aligned} &= (98 / 2) + 1 \\ &= (49) + 1 \\ &= 50 \text{ buah biji} \end{aligned}$$

Untuk setiap level penelusuran, jarak ketercapaian terhadap syarat cukup dapat dilihat pada tabel 5.2 berikut ini.

**Tabel 5.2 Aspek kecukupan untuk menang**

Jika ditelusur sedalam (N) Level	Keuntungan terbesar yang mungkin diperoleh (K)	Urutan langkah untuk mencapai K (U)  Ket : - a* = langkah mati di lubang a...	Minimal jumlah biji pada lumbung yang harus dimiliki untuk bisa menang sebanyak	Jarak keuntungan terhadap syarat kecukupan untuk menang	Belum/ Sudah dapat memprediksi kemenangan
1	9	alternatif 1 : a2→a2*, alternatif 2 : a3→a3*, alternatif 3 : a4→a4*, alternatif 4 : a5→a5*, alternatif 5 : a6→a6*, alternatif 6 : a7→a7*	50	-41	belum
2	18	a1→LA; a6→a5*	50	-28	belum
3	27	a1→LA; a3→LA; a2→a3*	50	-23	belum
4	38	a1→LA; a2→LA; a7→LA; a1→a1*	50	-12	belum
5	38	a1→LA; a2→LA; a7→LA; a1→a1*	50	-12	belum
6	38	a1→LA; a2→LA; a7→LA; a1→a1*	50	-12	belum
7	45	a1→LA; a2→LA; a5→LA; a2→LA; a1→LA; a5→LA; a1→a1*	50	-5	belum
8	46	a1→LA; a2→LA; a7→LA; a7→LA; a1→LA; a3→LA; a2→LA; a3→a5*	50	-4	belum
9	53	a1→LA; a2→LA; a7→LA; a7→LA; a1→LA; a3→LA; a1→LA; a4→LA; a7→a1*	50	+3	sudah
10	53	a1→LA; a2→LA; a7→LA; a7→LA; a1→LA;	50	+3	sudah



		a3→LA; a1→LA; a4→LA; a7→a1*			
11	59	a1→LA; a7→LA; a5→LA; a5→LA; a7→LA; a5→LA; a1→LA; a6→LA; a5→LA; a5→LA; a3→LA;	50	+9	sudah
12	63	a1→LA; a7→LA; a5→LA; a3→LA; a7→LA; a7→LA; a6→LA; a5→LA; a6→LA; a2→LA; a1→LA; a6→a6*	50	+13	sudah
13	69	a1→LA; a7→LA; a4→LA; a1→LA; a6→LA; a4→LA; a4→LA; a6→LA; a3→LA; a7→LA; a7→LA; a5→LA; a7→a2*	50	+19	sudah
14	69	a1→LA; a7→LA; a4→LA; a1→LA; a6→LA; a4→LA; a4→LA; a6→LA; a3→LA; a7→LA; a7→LA; a5→LA; a7→a2*	50	+19	sudah
15	75	a1→LA; a2→LA; a5→LA; a2→LA; a1→LA; a5→LA; a2→LA; a7→LA; a2→LA; a4→LA; a5→LA; a7→LA; a5→LA; a2→LA; a7→a5*	50	+25	sudah
16	76	a1→LA; a2→LA; a7→LA; a7→LA; a1→LA; a3→LA; a1→LA; a3→LA; a1→LA; a5→LA; a6→LA; a6→LA; a7→LA; a5→LA; a7→LA; a1→a1*	50	+26	sudah
17	77	a1→LA; a2→LA; a7→LA; a7→LA; a1→LA; a3→LA; a1→LA; a3→LA; a1→LA; a5→LA; a6→LA; a6→LA; a7→LA; a5→LA; a7→LA; a5→LA; a7→a1*	50	+27	sudah

Pada Tabel 5.2 di atas, terlihat bahwa untuk dapat menang mulai sejak awal permainan, pemain harus menelusuri DLS paling sedikit sedalam 9 level. Penelusuran yang lebih dangkal dari itu belum bisa menjamin bahwa pemain akan menang.

Atas dasar syarat kecukupan di atas, pada penelitian ini kedalaman dibatasi hingga kedalaman 17 level. Kedalaman 17 ditentukan dengan asumsi bahwa kemungkinan menang dapat dipenuhi dan kebutuhan memori juga masih dapat dipenuhi. Di sisi teknis, adanya pembatasan ini juga sangat menguntungkan bagi perangkat keras yang memiliki kapasitas memori yang terbatas karena tidak perlu membutuhkan penyimpanan stack yang terlalu banyak.



## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Dari penelitian ini dapat disimpulkan beberapa hal sebagai berikut.

- a. Circular linked list mampu memberikan fleksibilitas bagi aplikasi game congklak dalam mengelola spanning tree akibat dari proses pembangkitan langkah permainan. Fleksibilitas circular linked list sangat dirasakan terutama karena tidak diketahuinya banyaknya langkah-langkah permainan (beserta turunannya) yang akan terbangkitkan akibat upaya pencarian solusi.
- b. Pada setiap kedalaman level yang berbeda dari DLS, dapat ditemukan nilai optimasi yang berbeda pula. Artinya nilai keuntungan tertinggi permainan yang dicari dengan kedalaman level tertentu belum tentu menjadi nilai tertinggi pada level yang lain.
- c. Penelusuran DLS yang dilakukan semakin dalam akan memberikan prediksi kemenangan yang semakin baik, namun akan berakibat pada semakin besarnya memori linkedlist yang dibutuhkan. Dalam hal ini circular linkedlist memberikan sifat yang lentur terhadap kebutuhan memori.

#### **6.2 Saran**

Game congklak digital masih perlu dikembangkan untuk dapat memberikan efek visual maupun audio yang menarik seperti permainan congklak yang sesungguhnya.



## DAFTAR PUSTAKA

- Agung Budi Prasetyo, 1996, *Visualisasi Eliminasi Gaus pada Sparse Matrix dengan Partial Pivoting dan Complete Pivoting*, Universitas Kristen Duta Wacana, Skripsi.
- Anita Desiani & Muhammad Arhami, *Konsep Kecerdasan Buatan*, Penerbit Andi Bois, 20--, "*Permainan Congklak*", <http://www.bangbois.co.cc>.
- Dadan Umar Dani, *Komputerisasi Pengambilan Keputusan*, Elex Media Komputindo
- Expat Web Site Association Jakarta, Indonesia, *Instructions for How to Play Congklak*, diunduh dari alamat <http://www.expat.or.id/info/congklakinstructions.html>, pada tanggal 26 Januari, pukul 13'36 wib.
- Forchee, J., 2006, *Culture & Customs of Indonesia*, Greenwood Press, 88 Post Road West, Westport, CT 06881, ISSN 1097-0738, ISBN 0-313-33339-4, halaman 177.
- Insap Santosa,Ir, 1992, *Struktur Data menggunakan Turbo Pascal 6.0*, Andi Publisher, Yogyakarta.
- Kadir,A., 2009, *Dasar Pemrograman Java 2*, Penerbit ANDI, Yogyakarta
- Kustanto, C., *Penerapan Algoritma BFS dan DFS pada FTP Search Engine for ITB Network*, Institut Teknologi Bandung, Jurnal Penelitian.
- L.N.Harnaningrum, 2011, *Struktur Data dengan Java*, Graha Ilmu, Yogyakarta.
- L.N.Harnaningrum, *Algoritma dan Pemrograman menggunakan Java*, Graha Ilmu
- Matius Soesilo Wijono, 2005, *Java 2 SE Dengan JBuilder*, Penerbit Andi, Yogyakarta.
- Matius Soesilo Wijono, dkk , *Pemrograman GUI Swing Java dengan NetBeans 5* Penerbit Andi
- Munawar, *Pemodelan Visual dengan UML* , Graha Ilmu
- Philipus, S.M., *Making Applications Mobile Game Congklak Use Network Bluetooth*,Universitas Gunadarma, Jurnal Penelitian.
- Pusat Kajian Mainan dan Permainan Rakyat Komunitas Hong, *Upaya Melestarikan Permainan Rakyat -*, Koran Jakarta , 09 Maret 2013.
- Rohmah,H.H.,2012,*Bergesernya Dan Tersisihnya Permainan Tradisional Oleh Permainan Modern Sekarang Ini*, Kompas, Desember 2012.
- Sri Hartati, 2007, *Pemrograman GUI Swing Java*, Penerbit Andi, Yogyakarta.
- Sri Hartati, 2007, *Pemrograman Java Servlet dan JSP*, Penerbit Andi, Yogyakarta.
- Sri Hartati & Sari Iswanti, *Sistem Pakar*, Graha Ilmu
- Sri Kusumadewi, *Artificial Intelligence* Graha Ilmu.
- Sulun, H.S., 2010, *Pembangkit Teka-Teki Silang Dengan Algoritma Backtracking Serta Aplikasi Permainannya Yang Berbasis Web*, Institut Teknologi Bandung, Jurnal Penelitian.
- Suyanto, S.T.,M.Sc, *Soft Computing membangun mesin ber-IQ Tinggi*, Penerbit Informatika
- Dr.Suyoto, *Intelegensi Buatan*, Gava media
- Theresia, G., 2011, *Implementasi Algoritma DLS Pada Permainan Peg Solitaire*, Universitas Kristen Duta Wacana, Jurnal Penelitian.



- Tyas, P.M.A., 2010, *Implementasi Algoritma Backtracking Dengan Menggunakan Metode DFS Pada Penyelesaian Traveling Salesman Problem Suatu Digraph*, Jurnal Penelitian.
- Theresia, G., 2011, *Implementasi Algoritma DLS Pada Permainan Peg Solitaire*, Universitas Kristen Duta Wacana, Jurnal Penelitian.
- Tyas, P.M.A., 2010, *Implementasi Algoritma Backtracking Dengan Menggunakan Metode DFS Pada Penyelesaian Traveling Salesman Problem Suatu Digraph*, Jurnal Penelitian.
- Utomo, Y.W., 2011, *Konten Lokal di Game Online Masih Minim*, Kompas, 7 Juli 2011.
- Winarno Surakhmad, *Pengantar Penelitian Ilmiah*, Tarsito Bandung



## LAMPIRAN 1: RINCIAN BIAYA PENELITIAN

### Rincian biaya penelitian:

a. Honorarium	Rp. 200.000,00
b. Bahan dan Peralatan penelitian	Rp. 2.050.000,00
i. Pengangkutan barang	Rp. 150.000,00
ii. Buku	Rp. 200.000,00
iii. Jeketron	Rp. 450.000,00
iv. Alat & Bahan Penelitian	Rp. 250.000,00
c. Biaya lain-lain	
i. Pengisian Isotop	Rp. 150.000,00
d. Total biaya	Rp. 3.050.000,00

### Rangkuman biaya penelitian:

a. Honorarium	Rp. 200.000,00
b. Bahan dan Peralatan penelitian	Rp. 2.050.000,00
c. Biaya lain-lain	Rp. 150.000,00
d. Total biaya	Rp. 3.050.000,00



## LAMPIRAN 1 : RINCIAN BIAYA PENELITIAN

### Rincian biaya penelitian :

a. Honorarium		Rp. 800.000,00
b. Bahan dan Peralatan penelitian		Rp.2.050.000,00
i. Pengumpulan jurnal	Rp. 150.000,00	
ii. Buku	Rp.1.000.000,00	
iii. Internet	Rp. 150.000,00	
iv. Alat & bahan Penelitian	Rp. 750.000,00	
c. Biaya lain-lain :		
i. Penulisan laporan		<u>Rp. 150.000,00</u>
d. Total biaya		<u>Rp.3.000.000,00</u>

### Rekapitulasi biaya penelitian :

a. Honorarium	Rp. 800.000,00
b. Bahan dan Peralatan penelitian	Rp.2.050.000,00
c. Biaya lain-lain :	<u>Rp. 150.000,00</u>
d. Total biaya	<u>Rp.3.000.000,00</u>



**SURAT KEPUTUSAN**

**Ketua Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM Yogyakarta**  
**Nomor : L.05.1/1551/XI/2011**

**Tentang**

**PEMBERIAN DANA PENELITIAN BAGI PARA TENAGA PENGAJAR TETAP**  
**STMIK AKAKOM YOGYAKARTA SEMESTER GANJIL TAHUN ANGGARAN 2011/2012**

- Menimbang** : 1. bahwa dalam rangka pelaksanaan Tridharma Perguruan Tinggi, setiap dosen tetap diwajibkan untuk melaksanakan penelitian yang merupakan salah satu tugas tenaga pengajar dalam rangka memperoleh angka kredit untuk menduduki Jabatan Fungsional tenaga pengajar berdasarkan Keputusan Menteri Negara Koordinator Bidang Pengawasan Pembangunan Dan Pendayagunaan Aparatur Negara Nomor: 38/ Kep/MK.WASPAN/8/1999, tanggal 24 Agustus 1999.
2. bahwa pemberian dana penelitian dan pengembangan pada Semester Ganjil 2011/2012 melalui Pusat Penelitian dan Pengembangan dan PPM.
3. bahwa demi tertib administrasi dalam pemberian dana penelitian tersebut perlu ditetapkan dengan surat keputusan Ketua STMIK AKAKOM.
- Mengingat** : 1. Statuta Sekolah Tinggi Manajemen Informatika Dan Komputer AKAKOM.
2. Peraturan Yayasan Pendidikan Widya Bakti Nomor : 01 Tahun 1993.
3. Peraturan Yayasan Pendidikan Widya Bakti Nomor : 02 Tahun 1993.
- Memperhatikan** : 1. Peraturan Pemerintah Nomor 60 Tahun 1999
2. Keputusan Direktur Jendral Pendidikan Tinggi :
- a. Nomor 141 / D/ Q/ 1989 tanggal 26 Januari 1989.
- b. Nomor 262 /DIKTI /Kep / 1992 tanggal 8 Juni 1992
3. Keputusan Menteri Pendidikan Nasional Republik Indonesia Nomor : 074/U/Nomor : 074/U/2000
4. Undang-undang Republik Indonesia Nomor : 20 Tahun 2003

**MEMUTUSKAN**

Pertama;

Pemberian dana penelitian dan pengembangan pada Semester Ganjil 2011/2012, Peneliti tersebut di bawah ini disertai tugas untuk melaksanakan penelitian dengan judul :

No.	Nama Peneliti	Judul Penelitian	Biaya Penelitian
2	Agung Budi Prasetyo. S.Kom.,M.Kom.	Implementasi Cicular Linked Pada Pembuatan Game Permainan Tradisional Congklak	Rp. 3.000.000,00

Kedua;

Pembayaran dana penelitian akan dibayarkan sebanyak tiga kali, dengan rincian sebagai berikut :

- a. 30 % dari biaya yang ditetapkan, dibayarkan pada saat proposal penelitian disetujui dan diterbitkannya surat keputusan
- b. 30 % dari biaya yang ditetapkan, dibayarkan pada saat penyerahan draf laporan dan draf naskah publikasi



- c. 40 % dari biaya yang ditetapkan, dibayarkan pada saat yang bersangkutan menyampaikan laporan hasil penelitian dan naskah publikasi ( *hard copy* dan *soft copy* )

Ketiga;

Jangka waktu pelaksanaan kegiatan penelitian tersebut selama 6 bulan, terhitung mulai tanggal 1 Desember 2011 sampai dengan 1 Juni 2012

Keempat;

Apabila peneliti tidak dapat melaksanakan dan menyelesaikan penelitian tepat waktu, maka harus mengembalikan dana yang telah diterima kepada Kepala Puslitbang dan PPM, yang selanjutnya akan dikembalikan ke Kas STMIK AKAKOM Yogyakarta

Kelima;

Apabila dikemudian hari terbukti bahwa judul penelitian yang disetujui tersebut dijumpai adanya indikasi duplikasi dengan penelitian lain, dan/atau diperoleh indikasi ketidakjujuran dan itikad kurang baik yang tidak sesuai dengan kaidah ilmiah, maka penelitian tersebut dinyatakan batal dan peneliti wajib mengembalikan seluruh dana penelitian yang telah diterima.

Keenam;

Peneliti wajib menyampaikan laporan hasil penelitian yang telah disahkan oleh Ketua STMIK AKAKOM, sebanyak 3 (tiga) eksemplar yang akan diarsipkan pada Kepala Pusat Penelitian dan Pengembangan, Kepala Subbag. Kepegawaian dan Perpustakaan STMIK AKAKOM Yogyakarta.

Ketujuh;

Segala biaya yang berkenaan dengan pelaksanaan surat keputusan ini dibebankan kepada anggaran Pusat Penelitian dan Pengembangan dan PPM, tahun anggaran 2011/2012, kode 07.6/K.

Kedelapan;

Apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini, akan diadakan perbaikan sebagaimana mestinya.

Ditetapkan di : Yogyakarta

Pada tanggal : 17 November 2011

Ketua,

Sigit Anggoro, S.T., M.T.

NPI. 971104.

Tembusan disampaikan kepada :

1. Kepala Pusat Penelitian dan Pengembangan dan PPM
2. Arsip